DISI - Via Sommarive 14 - 38123 Povo - Trento (Italy)
http://www.disi.unitn.it

REPORT ON METHODS AND ALGORITHMS FOR BOOTSTRAPPING SEMANTIC WEB CONTENT FROM USER REPOSITORIES AND REACHING CONSENSUS ON THE USE OF SEMANTICS

Pierre Andrews and Juan Pane and Ilya Zaihrayeu

January 2010

Technical Report # DISI-10-009

INSEMTIVES
FP7-ICT-2007-3
Contract no.: 231181
www.insemtives.eu

INSEMTIVES

Deliverable <2.2.1>

Report on methods and algorithms for bootstrapping Semantic Web content from user repositories and reaching consensus on the use of semantics

Editor:	Pierre Andrews, DISI, UNITN
Deliverable nature:	<Report (R)>
Dissemination level: (Confidentiality)	<Public (PU)>
Contractual delivery date:	November 2nd, 2009
Actual delivery date:	
Suggested readers:	researchers in semantic web, information extraction and retrieval; developers of the bootstrapping toolkit in WP4
Version:	1.0
Total number of pages:	29
Keywords:	Bootstrapping | Semantic Web | Annotation | Semantic tag cloud | context | information extraction | lifecycle | cold start | annotation | metadata

Abstract

The INSEMTIVES projects explores two approaches to tackle the issue of missing semantic content in the semantic web: finding insentives to motivate the user to provide more annotation and minimising the cold start issue to provide enough critical mass of annotation to the users so they can see the benefits of semantic content. In this deliverable, we discuss solutions for reaching a critical mass of quality semantic annotations. We describe two techniques to solve this issue: a) automatic bootstrapping of annotations from the user's knowledge and the content of resources and b) consensus reaching techniques to ensure quality annotations that are understood and shared by most of the users of the semantic web.

Impressum

[Full project title] INSEMTIVES – Incentives for Semantics

[Short project title] INSEMTIVES

[WP2 – Models and Methods for the Creation and Usage of Lightweight, Structured Knowledge]

[Report on methods and algorithms for bootstrapping Semantic Web content from user repositories and reaching consensus on the use of semantics]

[Editor: Pierre Andrews, DISI, UNITN]

[Work-package leader: Ilya Zaihrayeu, DISI, UNITN]

[Estimation of PM spent on the Deliverable: 3.4]

Acknowledgement

The project is co-funded by the European Union, through the ICT Cooperation programme http://cordis.europa.eu/fp7/cooperation/home_en.html

Executive summary

The concept of semantic web has been around since 2001 when it was described in an article of the Scientific American [5]. However, eight years later, with a strong formal research behind us on the representation of semantics and how to use the semantic web, the actual use of this "new" web is minimal.

One of the issues that is encountered is the one of cold start. The semantic web will only work if there is enough semantic content available on the network to feed and leverage the semantic services that have already been researched. This content is still very costly to build and the lambda user does not yet see any reason to spend energy in creating such semantic content.

This is one of the multiple chicken and egg issues of the semantic web. If there is no semantic data available, the semantic services cannot work and the users cannot see the reason to provide such semantic data. One solution that we propose to explore in the INSEMTIVES project is to solve part of the cold start problem by:

1. automatically bootstrapping the semantic annotations of the users' resources by using the implicit semantics already contained in the content of these resources, but also in their context. In particular, the context provided by the users: what do they do with the resource, how did they create it and where do they store it.

2. helping the users to reuse a shared controlled vocabulary to semantically annotate the resources. By providing automatic and semi-automatic consensus reaching support techniques, we propose to make the diverse sets of terms used by the users to converge towards a shared controlled vocabulary where the meaning of these terms is described in formal semantic.

In this deliverable, we describe the outstanding issues drafted above, study the related work and existing solutions available in the state-of-the-art and propose solutions, combining the existing techniques with novel research to solve the cold start issue in the semantic web.

List of authors

Company	Author
<UNITN>	<Pierre Andrews>
<UNITN>	<Juan Pane>
<UNITN>	<Ilya Zaihrayeu>

Contents

List of Figures

Abbreviations

EXIF *EXchangeable Image file Format* is a standard for storing metadata about the creation of a photo in the header of image files.

IPTC *International Press Telecommunications Council* defined a standard file format for describing metadata (such as title, description, keywords) of a photo often referred to as IPTC.

Definitions

The Platform refers to the annotation storage platform developed by WP3 and described in D3.1 [38].

Document/Resources In this deliverable, we interchangeably use the terms "document" and "resources" for the resources that will be annotated and stored in the annotation platform and defined in D2.1.2 [7].

Annotation When we refer to an "annotation", we refer to a piece of metadata describing a resource and stored in the platform. When not specified, this refers to any annotation elements defined in D2.1.2 [7], controlled or uncontrolled.

Term as defined in D2.1.2 [7], A "term" is a non-empty finite sequence of characters. Normally, terms represent natural language words such as "sea", "bird", or "location".

Concept as defined in D2.1.2 [7], a "concept" is a node in a taxonomy formalising the semantics of terms.

Vocabulary When we refer to a "vocabulary", it refers to a set of "terms" used in all annotations available in the platform. In a "controlled vocabulary" these terms are linked to a taxonomy structure formally defining their semantic as defined in D2.1.2 [7]. In an "uncontrolled vocabulary" this set of terms refers to all sequence of characters used in the platform for *uncontrolled tag annotations* as defined in D2.1.2 [7]. When not specified, we refer to the union of the *controlled* and *uncontrolled* set of terms.

Feature A feature refers to an information describing a document. This feature can be an explicit metadata expressed in formal semantic or a "blackbox" feature with no concrete semantic when taken outside of the context of the resource and the algorithm that extracted this feature.

1 Introduction

In the previous deliverables from workpackage 2 (WP2), we have described the models for annotating resources with controlled and uncontrolled semantic metadata. The model that was defined in D2.1.2 [7] can store uncontrolled tags, which are simple textual strings or more complex controlled annotations that relate terms to concepts in a semantic taxonomy describing their meaning. The next step from this model is to get users to use it and populate its content. However, we face two main issues in doing this:

- creating semantic annotations is costly for the users and they do not yet see the benefits of it,
- in a shared annotation system, where users are free to provide the annotations they want, there is no guarantee that the vocabulary they use will converge and allow semantic services.

A solution to the first issue is explored in other workpackages of the INSENTIVES project to find incentives to motivate the user to provide more annotations. However, in this deliverable, we are interested in a different approach that will require minimum involvement of the users. We are proposing to use automatic methods to bootstrap the annotations of resources published by the users from their local computers. These techniques will extract semantic annotations from different sources of information: a) the context where the resource sits on the user's computer, b) the content of the resource and c) the knowledge of the user and the community. The related work is studied and we propose a novel solution in Section 3.

In Section 4, we propose a solution to tackle the second issue discussed above. We discuss the existing work and propose automatic and semi-automatic solutions to transform uncontrolled annotations into annotations linked to concepts in the controlled vocabulary.

In the next section, we start by analyzing the possible life cycle of annotations within the INSENTIVES annotation platform to illustrate the need for the bootstrapping of annotations and where the consensus reaching process will integrate in the annotation creation and maintenance cycle.

2 Annotation life cycle

In this section we describe a life cycle of resource annotations that consists of seven phases. The complete life cycle may include other phases, whereas in this section we describe only those which are further explored in the INSEMTIVES project. The life cycle is depicted in Figure 2 and described in the following paragraphs:

Phase I: Publishing. In this first step a resource is published on a network (arrow 1 in Figure 2). Without loss of generality we define *publishing* as a process by which the resource is made available in the network by means of assigning it a URL and a dereferencing mechanism that allows to retrieve a representation of the resource from the client machines on the network. This is a preliminary step that enables the annotation of the resource on the network;

Phase II: Bootstrapping. Once a resource has been published, the information about the context in which this resource resides in the user local repository is lost. However, this context information is a rich and subjective source of implicitly assigned metadata, manually or semi-automatically defined by the user that should not be lost. One typical example of a context is a (possibly) taxonomic organisation of personal photos into a hierarchy of folders in which folder names refer to periods in time, names of people, geographic places, and other content-related information. The goal of the second step of the life cycle, *bootstrapping*, is to preserve this implicit information that is codified in the context and in the content of the resource. Optimally, bootstrapping should take place right after publishing a resource which is then enriched with automatically extracted metadata by the bootstrapping process. Related approaches to bootstrapping and proposed solutions are discussed in Section 3 of this deliverable;

Phase III: Annotation. Published resources can be manually annotated by the users of the network. The process of annotation is the process in which the user assigns one or more annotation elements such as tags, attributes, or on of the other kinds defined in deliverable D2.1.2 [7]. As defined in [7], annotations can belong to two categories:

uncontrolled annotations (arrow 3 in Figure 2) are annotations that are *not* linked to concepts of a controlled vocabulary and only stored as free text strings with no formal semantics. The information retrieval tasks on uncontrolled annotations are normally reduced to the problem of computing string similarity between terms used in the query and those used in annotations. When using uncontrolled annotations, the user is free to provide an arbitrary text input that, however, may need to conform to application dependent rules (e.g., it must have no spaces);

controlled annotations (arrow 4 in Figure 2) are annotations that are linked to concepts of a controlled vocabulary (see [7] for details). The information retrieval tasks on controlled annotations can take advantage of the knowledge codified in the controlled vocabulary (e.g., expand query using synonymous terms) and can involve reasoning (e.g., about generality/specificity relationship between vocabulary concepts).

When using controlled annotations, the user uses terms and concepts from the vocabulary for the specification of tags, attribute names, relations, and other kinds of annotations (see D2.1.2 [7]). In principle, controlled annotations are key enablers of semantic services and, therefore, are the target annotation kind of the INSEMTIVES project.

Phase IV: Ontology Maturing via Consensus Reaching. The phase of ontology maturing represents the process by which uncontrolled annotations are evolved into controlled annotations following a consensus reaching process which is detailed in Section 4 of this deliverable (arrow 7 in Figure 2). In other words, ontology maturing is the process of enriching the controlled vocabulary with new terms and concepts that have been used as uncontrolled annotations by the users on the network. Ontology maturing can follow two main scenarios:

manual maturing (arrow 5 in Figure 2) is the scenario in which the user manually moves an uncontrolled annotation to the controlled vocabulary by specifying the necessary information about the newly added term and concepts such as synonymous terms, more general or more specific concepts. In this process, the user may receive suggestions coming from external knowledge bases such as

DBPedia[1] and the like (arrow 6 in Figure 2). Because it is performed by a human, the manual maturing process is expected to be of a relatively high quality;

automatic maturing (arrow 10 in Figure 2) is the scenario in which the system automatically enriches the controlled vocabulary thought the analysis of use of uncontrolled annotations by communities of users on the network. This process is directly related to the problem of reaching consensus which is detailed in Section 4 of this deliverable. Because it is performed automatically, the automatic maturing process is expected to be of a lower quality than the manual maturing process.

Phase V: Annotation evolution. Because the controlled vocabulary evolves in time, annotations (both controlled and uncontrolled) are subjects of evolution in time. Without loss of generality, we define the *annotation evolution* as a process in which links from controlled and uncontrolled annotations to resources are recomputed as the structure of the vocabulary changes. This problem is described in deliverable D2.2.2 [9] and herein we provide an example for the sake of clarity: consider a scenario in which a resource R_i was annotated with an uncontrolled annotation "palm" (arrow 9 in Figure 2), which, in a later moment in time, was added to the controlled vocabulary as a new term T_j and a new concept C_k which means "palm tree" (arrow 10 in Figure 2). The problem now is to decide whether the uncontrolled annotation of R_i should be re-mapped to the new concept C_k in the controlled vocabulary (arrow 11 in Figure 2) or if it should remain an uncontrolled annotation as it has a different (not yet formalised) meaning than "palm tree"[2].

Phase VI: Linking to external repositories. In this phase controlled annotations are mapped to entries in external repositories through the definition of links from the concepts of the controlled vocabulary (to which controlled annotations are mapped) to external entries and the definition of the semantic relations that hold between them (arrow 8 in Figure 2). These links can enable semantic services that cross the boundary of the INSEMTIVES platform. For example, these links can be defined by the user as part of the manual ontology maturing process (read above). The problem of linking to external repositories is described in deliverable D2.2.2 [9];

Phase VII: Use. This phase includes the use of annotations for various purposes such as search and navigation (arrow 12 in Figure 2). This is where the end user gets the return on investment in annotating resources in the network.

[1] http://dbpedia.org

[2] it could refer to "palm computer" or "palm of the hand".

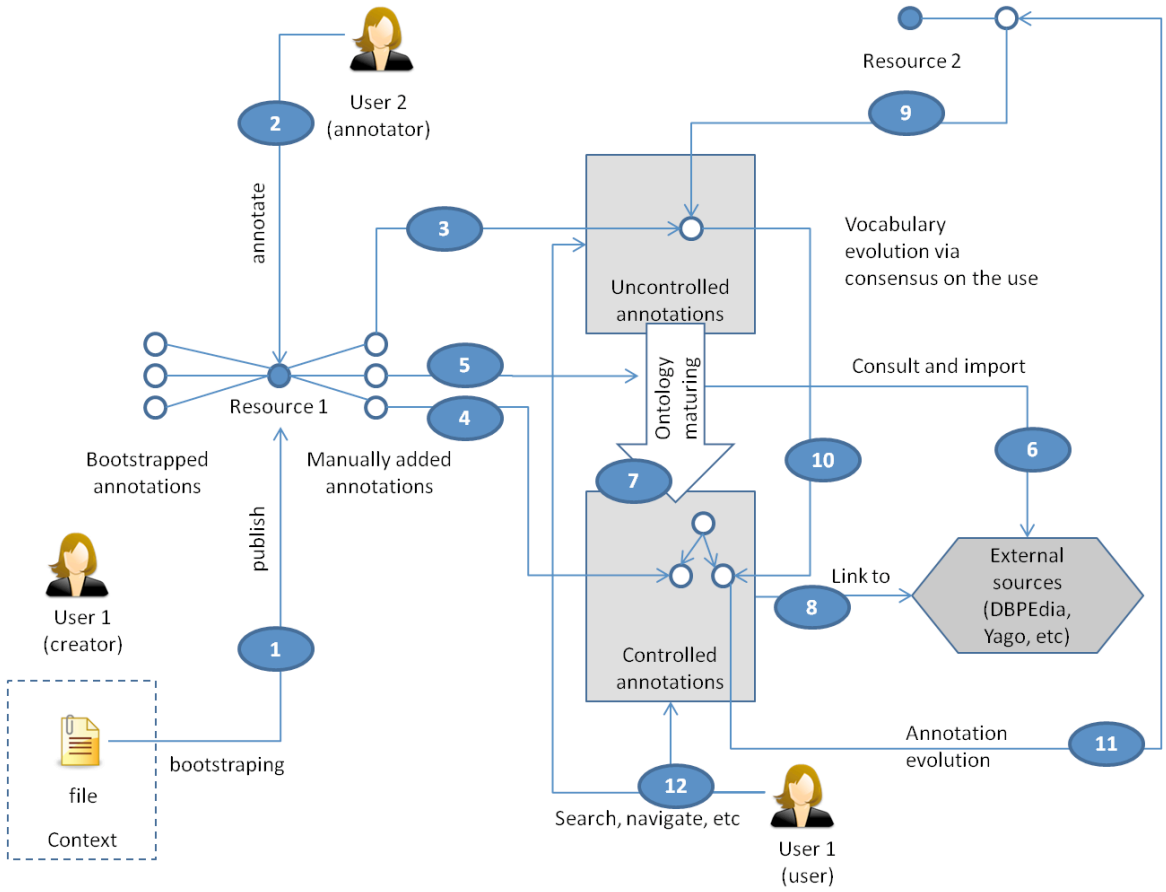

Figure 1: A life cycle of semantic annotations

3 Bootstrapping

3.1 Problem Statement

The INSEMTIVES project aim is to find solutions to increase the number of semantic annotations on resources on the web, the personal computer or within intranets. Increasing the mass of available annotation and its semantic complexity will allow for more accurate services (such as search for example). However, we believe that the cold start issue is a fourth chicken-and-egg problem in the semantic web in addition to the three technological ones identified in [21]: how to get the users to provide the semantic annotations when they do not yet see the benefits?

In the INSEMTIVES project, the research is investigating two complementary approaches to motivate the user to provide and use semantic annotations. The first approach aims at finding incentives to motivate the user, the second approach which is discussed in this section is to find methods for automatically *bootstrapping* annotations to show the users the benefits of having such annotations.

For instance, if we look at the Flickr[3] photo sharing service. In October 2009, they have passed the bar of the 4th billion photos hosted on this photo sharing website[4] which was one of the pioneering website to use a tagging system to let the users organise their documents. However, when studying a randomly selected sample of photos[5] from this website, we found that a majority (22.5%) do not use any annotation and if they are annotated, only a small number of annotation is used (21% have between 1 and 3 tags, see Figure 2).

Flickr uses easy to input free text tags, which are uncontrolled annotations, but users already have issues in providing enough of them to leverage services on their documents. However, many documents already have a number of implicit semantic annotations that are often lost when uploaded into an annotation platform but

[3] http://www.flickr.com
[4] http://blog.flickr.net/en/2009/10/12/4000000000/
[5] 114065 photos were studied

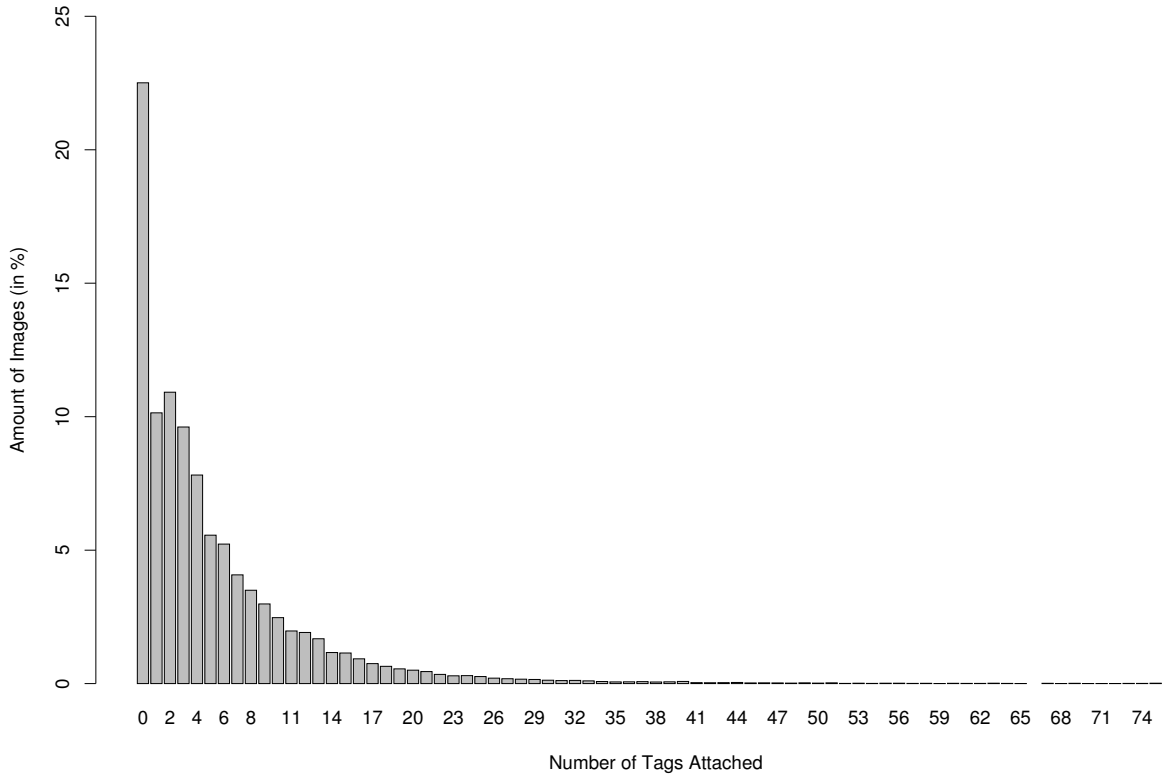

Figure 2: Use of Tags in a sample of 114065 Flickr photos. Bars represent the portion of the sample that are annotated with that number of tags (in percent of the sample size). A majority of photos have no tags (22.5%) or very few tags (21% have between 1 and 3 tags).

could be extracted with minimum involvement of the users to bootstrap the annotations. For example, a photo, before being uploaded on Flickr, is usually placed in the personal file system where the user will have created a personal folder organisation to find that photo again. In addition, that photo embeds a large amount of semantic information in its own metadata, through the EXIF and IPTC headers that keep information automatically generated by the camera about how and where the photo was taken. However, all these raw information contains noise and data that might not be suited to the web platform where it is uploaded – for example for accuracy reasons, or for privacy reasons.

Our hypothesis is that, before uploading a document to the shared annotation platform, we can automatically extract a number of raw information for that document that we can filter out to provide a bootstrapped annotation for that file. We have identified three different sources of raw information:

1. The file content,
2. The file context (for example where it is placed on the user's file system),
3. Similar files already annotated in the platform – i.e. user and community knowledge,

These three sources and how they can be combined to bootstrap a document's annotations are discussed in more details in the Section 3.3. In the next section, we present the related work that can be used to tackle the issue of bootstrapping annotations in the semantic web.

3.2 Related Work

Sheth et al.[37] describe the concept of *implicit semantics* that are contained in documents, the authors explain that in addition to the explicit semantic attached as controlled or uncontrolled annotations to a documents, the content of the file can provide additional information about its meaning (implicit semantic) that could be extracted and stored in *explicit* semantic controlled or uncontrolled annotations. However, the raw extracted information, as we illustrate later on, is not yet structured in sets of uncontrolled terms or formal concepts needed for the annotation within the annotation platform. The authors of [37] provide a review of possible

solutions in the existing research for creating controlled and uncontrolled annotations to bootstrap the semantic annotations of documents from the implicit semantics in its content.

A majority of the work present in the Information Extraction field of research focuses on the extraction of annotations from the content of textual documents. For instance SemTag [14] is a tool that can be trained to automatically annotate textual documents based on the annotations from a previously manually annotated training set. This approach is very similar to the task of named entity recognition and extraction (NER) in the Natural Language Processing field that is used by the KIM tool from OntoText [32]. While these tools try to link the mentions of terms in text to known entities in a knowledge base, Rajman et al. [33] take a different approach by trying to extract generic concepts representing the content of the document. Instead of extracting a direct Named Entity mention, the authors propose an algorithm to find common words in the text and map them to an ontology to then search for the most relevant subsuming concepts to represent these terms, thus extracting generic topics describing the content.

All these techniques work on textual content but there has also been research in extracting features from images, videos and other types of documents. [20] proposes a generic automatic annotation framework based on extractors that are specialised for different content type and are then combined to bootstrap the documents annotations.

These techniques are based purely on the content of files to automatically create new annotations. To the best of our knowledge there is not much research that focuses on using the context in which the document sits on the file system to find implicit semantics. Soules and Ganger [40] propose to use information from the file system to connect related files and propagate existing semantic annotations. In addition to using document similarity measures, they use the file system's access patterns to compute the relationship between a set of documents; they can send propagate existing annotation to related documents. In previous work developed by the University of Trento [3, 48] we propose to parse short natural language labels of the nodes of classifications, this could be used to extract the implicit semantics from a folder structure of a file system and use this to annotate files under that structure with the explicit semantics extracted from that context. Davis et al. [13] propose to use the live context where the document was created to create metadata describing this document. They use features from the context of a photo taken with a cellphone – such as when it was taken and to which cell tower the phone was connected – to infer metadata about the location where a photo was taken by comparing these features to already seen photos.

In comparison, leveraging the community knowledge to automatically increase the annotations of a user's file is already a well studied field of research in the context of tagging systems on the world wide web. For instance, Sood et al. [39] use similarity between textual documents (blog posts) to recommend tags for a new document based on a corpus of existing annotated documents. In this publication, the comparison of textual documents is based on a basic bag of words vector distance measure but it could be extended to more complex similarity measures already developed in the Information Retrieval field. Crandall et al. [12] use a similar technique to predict geographical annotation of images. They first characterise a set of geographic locations with the features of images already annotated with this location. They can then predict the location of a new image from the annotations (in this case tags) associated with the new image combined with image content features (through SIFT feature extraction [27]).

These annotation recommendation techniques are described as content specific, but the actual recommendation algorithm can be abstracted from the content feature extraction techniques to produce a generic recommendation platform. We discuss our approach to this generic annotation framework in the next section. Content independent techniques are already discussed by [19, 26, 16] who use metadata from documents (e.g. existing tags, descriptions, embedded metadata) in combination with the user's personal tag lexicon and similar documents tagged by the community to recommend a set of tags for the new document.

To the best of our knowledge, the existing work in automatically annotating documents from their content, context and the community knowledge often limits itself to simple structural complexity for the annotations (see deliverable D2.1.1 [8]), using *tags* as annotations or at best *relations* to entities annotated in part of the text (for example [32]). In addition, with the exception of the NER techniques, all the research studied extracts annotation without linking to a controlled vocabulary and just provide free text annotations with no explicit formal semantics. Our aim is to extract more complex structures from all the sources available, including attribute values and relations to entities and to be able to link the bootstrapped annotations to the controlled vocabulary provided by the platform.

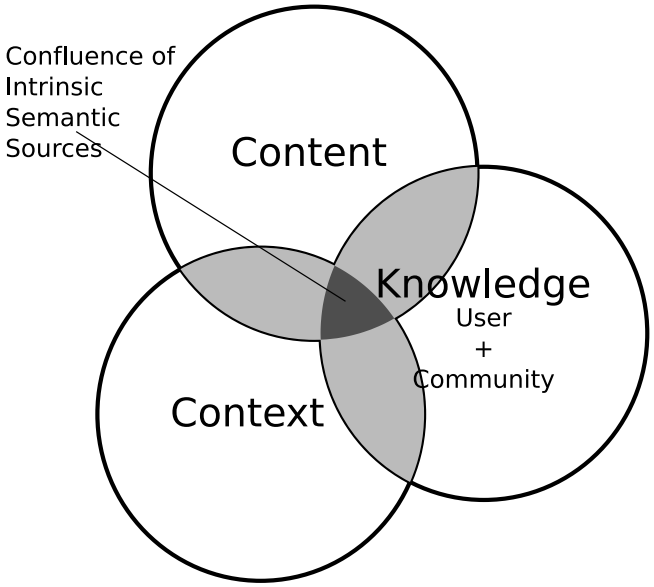

Figure 3: Confluence of Implicit Semantic Sources.

3.3 Proposed Solutions: Implicit Semantic Confluence

The existing work on automatically annotating documents presented in the previous section mainly focuses on extracting implicit semantic from the content of files or from the community knowledge. We believe that their is a third source of implicit semantics: the context of the document. The context can take many forms, it can be the local file system where the document is sitting (see [40] for an example of automatic annotation from the file system context), the context in which the file has been created (e.g. when, where [13]) or any other sources that is not directly related to the content of the document. We believe that the use of the context can provide very good quality implicit semantics and that it should be used as much as possible in annotation bootstrapping, in particular because such context is usually lost when uploading a document to a centralised platform while the content and community knowledge are always available.

In addition, to the best of our knowledge, the existing research in bootstrapping annotation to solve the cold start problem focuses only on one source of implicit knowledge and does not combine the different sources. We would like to introduce and develop the concept of *Implicit Semantics Sources Confluence* that will combine all the three sources of implicit semantics to extract and filter the most relevant explicit semantics. Our hypothesis is that by combining the different sources, we can find an overlap as illustrated in Figure 3 within the implicit semantics features coming from each source and extract more accurate explicit semantics to bootstrap the annotations.

In this section we discuss the generic architecture used for the confluence and the proposed techniques that can be used to implement such confluence engine. Figure 4 contains a high level view of the architecture that we are proposing. This architecture is divided in different modules:

CONTEXT EXTRACTION MODULES The context extraction module contains a number of sub-modules that are able to extract implicit semantics from the context and formalise them to a raw formal semantic that can then be used in the convergence engine to create the bootstrapped annotation.

The context from which the implicit semantics can be extracted depends on the type of document processed and the metadata it contains. For instance, an image file that has been created by a digital camera will embed EXIF headers with information on how the photo was taken, when it was taken and where (if the camera is combined with a GPS device). These metadata could be extracted to attributes for this file and relationship to other entities. A text document will not contain such metadata, and thus different context extractor might be needed to tailor the extraction process.

Example For document published to the platform from the user's local file system, the folder structure in which this file is stored locally provides a context from which we can extract implicit semantic information. For example, if we have a photo stored in the folder "2009-01-12 Tom" of the folder structure illustrated in

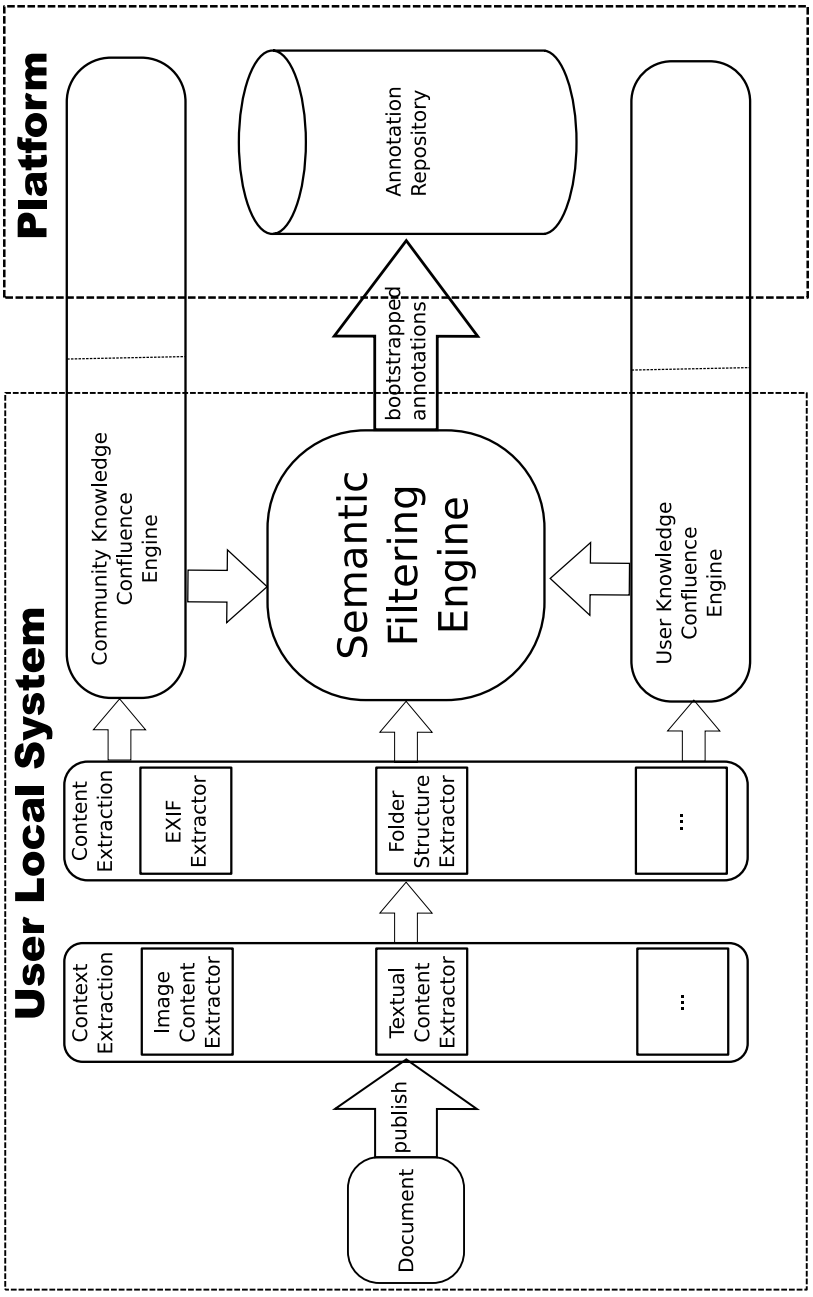

Figure 4: Framework for the extraction and confluence of implicit semantic sources.

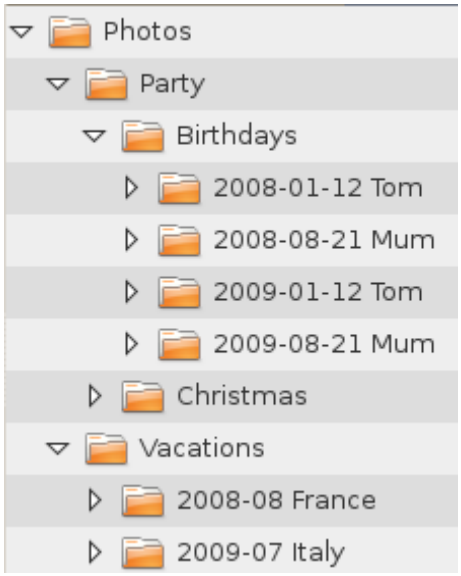

Figure 5: Example of a local file system classification

Figure 5, the context extractor could extract information about the date when the photo was taken, that it was taken at the "Birthday" of "Tom" and that this "Birthday" is a "Party". The information can be extracted using the natural language parsing technique that we presented in [3, 48] which propose a natural language parsing pipeline that is able to extract formal semantics from classifications. The algorithm can identify concepts mentions in short labels and disambiguate them to the corresponding concepts and entities in the controlled vocabulary.

CONTENT EXTRACTION MODULES Content extractors can extract a number of implicit semantic features from the content of documents. As for the context extraction, multiple sub-modules can coexist to process different type of contents or extraction techniques. The features extracted by the extraction module can already be explicit formal semantic representation but could also be implicit features describing an important aspect of the content. For instance, when processing textual content with a named entity extraction algorithm such as the one provided by KIM [32], explicit relations between terms in the text and entities from the controlled vocabulary can be extracted. However, when processing an image, the SIFT [27] invariant features extraction algorithm can be used to extract important features from the content. In the SIFT example, each keypoint of the image is described with a feature vector of 16 orientation histograms; how these features directly map to formal semantics – i.e. annotations in uncontrolled terms or controlled concepts on the platform – might not be known by the content extraction module. This issue may arise for any type of extraction module that is not directly able to extract lexical features, however, such mapping might be inferred later on by the knowledge confluence engines (see next sections) when comparing to other, already annotated, documents from the same user or from the community containing similar features.

KNOWLEDGE CONFLUENCE As mentioned above, the features extracted from the content and the context might not be already expressed in a formal representation corresponding to the uncontrolled terms or controlled concepts available in the platform. This might be because the content/context extraction modules extracted raw implicit features (such as the SIFT features discussed above) or because they extracted important lexical tokens without being able to disambiguate them.

For instance, in the folder structure example discussed above, the extractor can find that the "Birthday Party" if the one of "Tom", however, to which entity this term refers to cannot be disambiguated directly from the context. However, by matching this term, in combination with the birthday date, to the entities already used in the user's knowledge – i.e. all the uncontrolled or controlled annotations already used by the user in the platform –, the knowledge confluence module can find who "Tom" is in the user's context and can match it to a concrete entity in the platform vocabulary.

If the features extracted are implicit, such as the one provided by a SIFT extractor, these features cannot

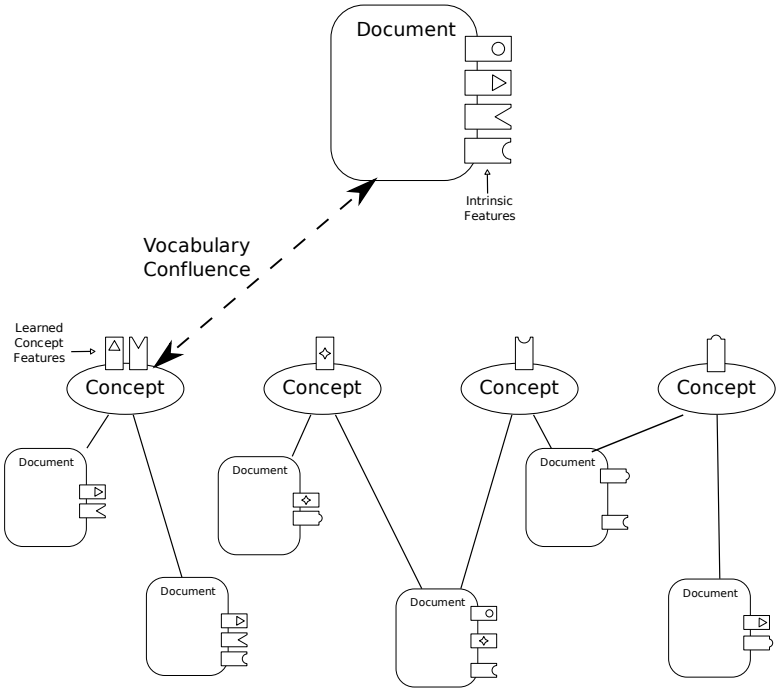

Figure 6: Matching Document Implicit Features with Concept from the Controlled Vocabulary Characterised by Attached Documents Common Features.

be semantically matched to annotation elements as defined in deliverable D2.1.1 [7] as we described in the previous paragraph. However, these features can be compared to similar features of other documents already annotated in the platform as illustrated in Figure 6. This can be compared to an automatic classification problem where each possible annotation values (for example a concept from the controlled vocabulary for a controlled tag annotation) represents a class and implicit features are used as classification features for the documents. A supervised classification algorithm can be trained on the existing annotated document instances in the platform and then used to classify new documents under the correct class/concept.

We could also see the problem as an automatic clustering problem where the number of classes are unknown. In this case, the clustering of existing documents according to their features would identify clusters of similar documents. Each cluster would then define a set of annotations common to all of these documents. The bootstrapping of the annotations of a new document would then require to find the cluster in which this document best fits according to its extracted features.

The *community knowledge confluence* module uses the same techniques as the *user knowledge confluence* module, but instead of aligning the features extracted from the context and the content to the user's own knowledge, this convergence module tries to find novel annotations from the knowledge (the combination of all controlled and uncontrolled annotations) used by other users of the platform. This process can find new annotations by comparing the extracted features to the most common features of the documents already annotated by the other users and how similar they are to the new document published by the user.

FILTERING ENGINE The different extraction and confluence modules provide a raw set of implicit annotation features that are mapped to existing annotations in the controlled vocabulary, in the set of uncontrolled terms, or represent new annotations introduced for this document. The role of the *Filtering Engine* is to select the most relevant annotations for this document, by comparing the different sources of implicit semantics and choosing the annotations that were extracted by most of the modules.

This can be done by ranking the raw annotations proposed by each modules according to the importance of use by the user, but also how much it is present in the current document and how much it is used by the community of users. In this way, the confluence modules avoid the introduction of too much noise in the annotation and the explosion of the polysemy of the terms in the platform by choosing the annotations that have already been validated as most important in the domain by the user and by the community.

3.4 Beyond the State of the Art

In this section we have proposed solutions to automatically bootstrap the semantic annotations of resources with new annotations that are linked to the controlled vocabulary of the platform. The introduced research builds on existing separate techniques that need to be combined to attain quality results.

In particular, we propose to develop new algorithm to formalise the labels found in the context of resources, based on work we have already started at the University of Trento [48, 3] but also to develop new algorithm, based on machine learning techniques such as clustering and classification, to align the extracted implicit semantics with the existing knowledge stored in the platform.

This is an important improvement on the current state-of-the-art as keeping the size of the set of uncontrolled terms and of the controlled vocabulary from exploding is important to avoid polysemy issues that could lower the accuracy of semantic services.

4 Consensus Reaching

4.1 Problem Statement

As from deliverable D2.1.2 [7], a controlled vocabulary is a key enabler of semantic annotations and services within the INSEMTIVES platform. This vocabulary can be constructed by experts but, as it follows from the requirements reported in [7], there must be the possibility for ordinary platform users to extend the vocabulary with new terms and relations. However, because this enrichment process is largely distributed and involves single individuals at a time, it is hard, if not impossible, to guarantee that all the users will extend the vocabulary in a uniform and consistent way, avoiding duplicate and redundant vocabulary entries, incorrect information, and other problems. Apart from this, many users will not be concerned about enriching and maintaining the controlled vocabulary and will use uncontrolled annotations (see [7]), at least, until they see an added value for doing it. However, in order to bridge the gap between uncontrolled and controlled annotations in the quality of services (QoS) and to ensure a graceful improvement of the QoS as more annotations (of any kind) are added, the system should support the (semi) automatic enrichment of the controlled vocabulary by computing its new elements from existing uncontrolled annotations. The key challenge is to find effective and user-friendly means to combine the two processes (i.e., user-driven and automatic vocabulary construction) such that the controlled vocabulary is kept constantly evolving in an as consistent and error-free manner as possible. The ultimate goal of this challenge is to ensure that the vocabulary is used by different users consistently, i.e., that different users use the same vocabulary elements for the same purpose(s).

Consider Figure 4.1 in which we show the number of (re)use of tags in a sample of 481,743 different tags used on 13,160,954 photos from Flickr. We can see that a majority of the tags (almost 62%) are used only once on the whole set of 13 million photos. Theoretically, all these 481743 tags could all be semantically different, and all the 13 million photos of a different subject, but the probability of this is very low. Relating these tags through the synonymy and more/less general links and helping users reuse them in a consistent manner would help avoid such problems as low correctness and completeness of search results brought by the synonymy, polysemy, and semantic gap problems of natural language (see deliverable D2.1.1 [8] for details).

In order to further exemplify the problem statement, below we provide an incomplete list of scenarios which illustrate the problem:

- with the growing popularity of the Google search engine, users started to use the uncontrolled annotation “to google” interchangeably with the controlled annotation “to search”. However, because “to google” is not part of the controlled vocabulary and, as the result, the fact that both terms can be treated as synonyms is saved nowhere, search results for “to search” do not contain resources annotated with “to google”, and vice versa. The system should provide a way to compute the synonymy link (e.g., thought the co-occurrence analysis) and add the new term to the controlled vocabulary;

- when annotating resources that represent cars, user U1 normally used the uncontrolled term “car” and user U2 normally used the uncontrolled term “automobile”. However, when user U2 searches for resources annotated with “automobile”, she does not find those annotated with “car” even if those resources are relevant to her query. With more users using these two terms interchangeably, the system can compute the fact that these two terms mean the same thing, create a new linguistic concept in the controlled vocabulary and attach these two terms to it. From now on, users annotating resources with the term “car” (or “automobile”) can link their annotation to the newly created concept in order to help them (and other users) find these resources when using the term “automobile” (or “car”) for searching;

- when annotating resources that represent material on the Java programming language, many users used the uncontrolled term “Java”; when annotating photos of the Java island, many users used the same uncontrolled term, “Java”. Because there is no syntactical difference between the two terms, searches for “java” return both, resources related to the programming language and to the island. However, if the system could compute the fact that the same term has different meanings, create entries for these meanings in the controlled vocabulary, re-map the existing uncontrolled annotations to the created entries, and let users use them for the annotation and search, then the above described problem would be solved to a certain extent;

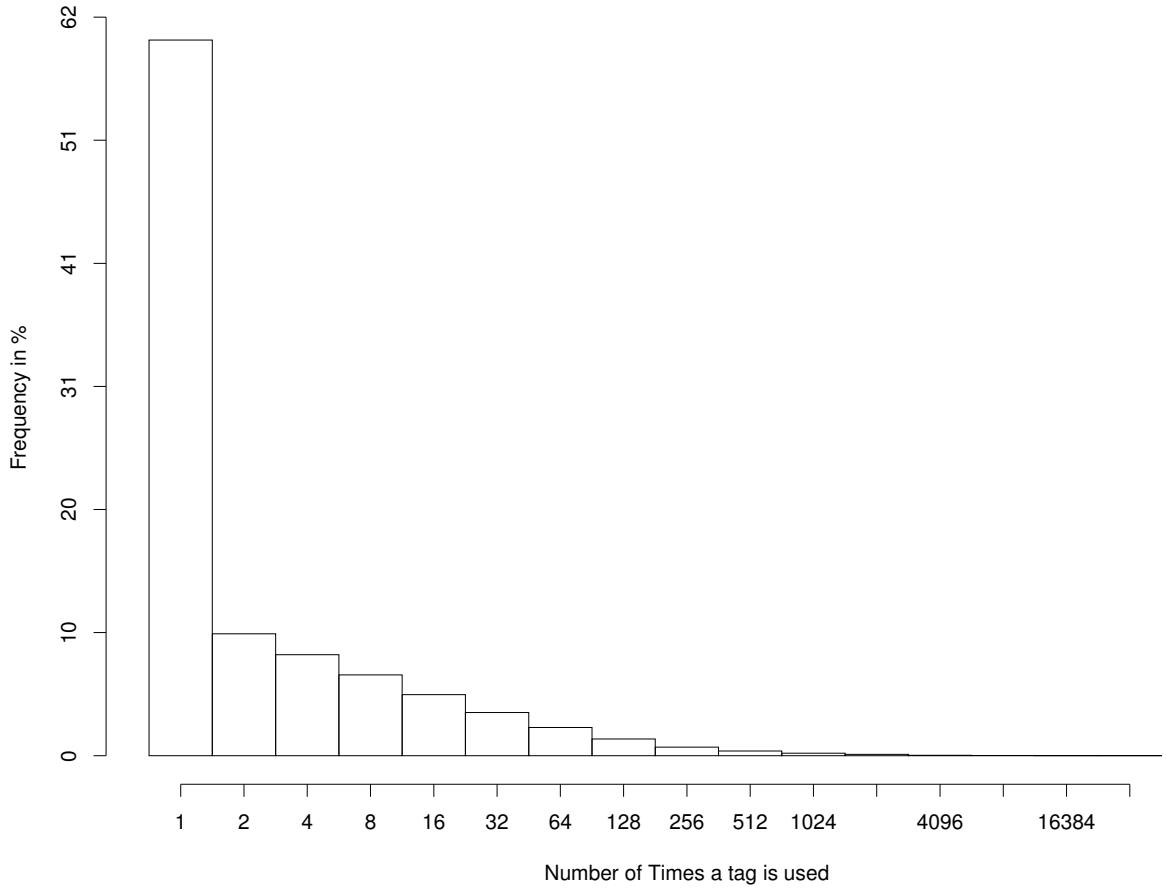

Figure 7: The number of (re)use of tags in a sample of 481,743 different tags used on 13,160,954 photos from Flickr.

- when describing the term "Java" (the island) many users linked it to a more general term "Island" while some other users erroneously linked it to the term "Programming Language". This incorrect link may lead to erroneous query results and, therefore, should be detected and deleted by the system automatically or with the help of system's users (who can report an erroneous link);
- when describing the term "Java" (the island) many users linked it to a more general term "Island" while some other users linked it to the term "Land" which is more general than the term "Island". While not being erroneous, this link is redundant and, therefore, complicates the understanding of the vocabulary structure for the system's users. The system should be able to detect and remove this link automatically or with the help of system's users (who can report a redundant link);
- as the controlled vocabulary changes with time, it may evolve into a state in which there are two or more terms, manually added by the users or computed automatically by the system, which mean the same thing and, therefore, are used in controlled annotations for the annotation of the same or similar resources. In a sense, it is the same problem as the one related to uncontrolled annotations (recall the example about terms "car" and "automobile"), but brought at the level of the controlled vocabulary. Therefore, it leads to the same problem of incomplete results and needs to be (semi)automatically resolved by the system. Differently from the example of uncontrolled annotations, the solution to this problem will require merging of relations that are defined for the two controlled terms in the vocabulary.

4.2 Related Work

The first notion we need to clarify is: what do we mean when we talk about consensus?. The work presented in [35] gives an overview of different perspectives about consensus for decision making. For the purposes of the present document we consider the following definition to be the most suitable: "*A decision-making process in*

which all parties involved explicitly agree to the final decision. Consensus decision making does not mean that all parties are completely satisfied with the final outcome, but that the decision is acceptable to all because no one feels that his or her vital interests or values are violated by it" [4]. In the context of our work the *parties* are the annotators, the *involvement* refers to the process of annotating a resource, and the *final decision* is the annotation itself.

Reaching a consensus on the use of semantics through the alignment and ***convergence*** *of vocabularies means that all the same applied annotations have the same meaning when applied to the same resource*. For example, if two or more people use the term "java" to annotate a given resource, then all of them have the same personal understanding of what "java" means. The issue is how to extract these understandings (semantics) that are encoded in the mind (personal knowledge) of each user, in such a way that the computer can later exploit this knowledge; this is where semantic web comes into play [5].

There are several approaches for modelling the evolution of knowledge [6] [42] [17] [36]. In general, the underlying assumption is that a group of people agree (there is a consensus) that there is a new term that has to be added to a form of controlled vocabulary. Different approaches focus on different kinds of groups of people, for example, in digital libraries [30] and thesaurus [29] the group is a small predefined set of experts; in more recent proposals [17] [36] the group is larger, exploiting the collaboration of communities of users. The work on emergent semantics [1] presents some principles and conditions under which a bottom-up approach of vocabulary evolution could take place. The work does not present a model, but is serves as reference for when to consider that emergent semantics could take place.

Braun et. al [6] proposed an "Ontology maturing" process based on a collaborative approach for evolving an ontology. The work defines the maturing process in four steps:

1. Emergence of ideas: new terms are entered, typically using free-text tags or terms (uncontrolled annotations).

2. Consolidation in communities: these new terms are reused by a community of users. This reuse suggests that all the members share the same understanding (semantic) for the term.

3. Formalization: Once the system recognizes a new commonly used term it tries to extract a hierarchical relation between the new term and the existing concepts in the ontology (controlled vocabulary).

4. Axiomatization: In this step the system should try to infer more domain specific semantics to extract axioms to add them in the ontology, allowing reasoning on top of this new knowledge.

Similarly, [11] present an algorithmic approach to address the vocabulary problem (homonyms, synonyms, ...) in collaborative environments. The author proposes three general steps to be followed: **i**. (common) vocabulary identification, **ii**. linking similar vocabularies (via term frequency and clustering techniques) **iii**. traversing the concept space (browsing the clusters using similar links). Although both works [6] [11] propose generic models and then test their models in use cases, those models could be considered as general steps to be followed in order to extract common vocabularies in collaborative settings. In the present document we consider steps 1, 2 and 3 from [6] presented above to be the general steps to be followed.

In [17] the authors propose a community-driven ontology evolution by allowing users to collaboratively edit classes of an a-priori existing ontology via tagging like mechanisms. The work proposes a rating system to evaluate consensus where the most generally accepted changes in the ontology will be rated top in the ranking. While the authors claim that the editing activities are made simpler for the users via the tagging-like mechanisms, it is still assumed that the users have at least some basic notions about ontologies. This assumption presents a constraint on the number of users that will be able to contribute in such model. In [46] the authors also propose a similar mechanism to exchange knowledge based on a voting-like mechanism to check whether interacting agents share the same understanding of a given vocabulary.

In [42] the authors present a model for tagging resources where the resources could also be the tags, therefore allowing to tag the tags, in a form of meta-tagging approach. The model also considers a mechanism of *appreciation* and *depreciation* of tags where users can rank (with + or -) other users' tags. This annotation model is very similar to our model described in [7] but the fundamental difference is that the work in [42] assumes no previously shared controlled vocabulary, and that the ontology will basically emerge from the meta-tagging process. This assumption poses an issue on bootstrapping and usability of the system, considering that

an initial mass of content should be initially present in the system in order to make it useful from the beginning (see Section 3).

In contrast to the previously cited work, where users manually evolve the ontology, there are several approaches [28] [23] that try to extract semantics from free-text tags (or folksonomies [45]) applied to resources. In [28] Mika proposed a model to extract lightweight ontologies using an Actor-Concept-Instance model similar to our defined model in D2.1.2[7]. In [23] the authors proposed an algorithm for extracting a navigable hierarchical taxonomy of tags based on cosine similarity of vectors. For a more detailed list of automatic extraction of hierarchical relations from free-text tags, please refer to [17].

Sen et al. [36] showed how using human computer interaction techniques and smart design of user interfaces could help the process of convergence of vocabularies. The authors experimented with several configurations of tagging systems, where users could see/not-see the tags applied to the same resource by other members of the community. The results obtained showed that community tags influenced the tags applied by single users, and that the number of tags applications also increased in the experiments where users could see community tags. Considering factual versus personal tag applications, in the experiments where users could see community tags, there were more factual tags. Another important finding of this work is the result of the survey; they found that users did not apply tags in the experiments where they were not shown community tags "*because they could not think of any tags*". This suggests that showing tags is one way for system designers to encourage more people to use tags.

Pasant and Laublet [31] proposed a model called MOAT where the underlying controlled vocabulary is aligned with external ontological resources following the Linked Data [47] principles. The idea is to link the tags applied by the users to meanings given by external ontological resources. The model defines that only after the user had saved the tags the system will look for the related meaning; making this procedure prone to problems like variation in plurals, part of speech, and others [18]. Furthermore, the model requires another interaction with the users for asking them to choose the correct meaning for the applied tag. This last step could be avoided if the tagging model queries the server while the user types the tag, avoiding vocabulary problems and saving one extra interaction with the user.

Samuels and Drake [34] present a study of convergence and divergence in *biological communities* and state that the convergence or divergence of the community to a stable state normally depends on the level of granularity with which the components are studied. In the scope of this document we could consider that the elements are the vocabularies and the resources, and the relation existing between them are the annotations, therefore, according to the level of specificity with which users annotate the content, we could achieve or not convergence, i.e., it is very difficult to achieve convergence in the use of vocabulary if different users apply different levels of specificity when tagging (consider a tag application of "dog" versus "poodle" for a picture). They also state that the "*nature would be intolerably regular if all communities converged to a single solution or attractor*"; in the context of the present document it means that there will be cases in which the concept will not converge (or it will be very hard to achieve convergence via automatic processes, and users will be needed to aid the process). Even more, the existence of divergence for a single term applied to several resources, means that this term has different meanings.

4.3 Proposed Solutions

The main goal of the proposed solution is to reach an agreement on the use of the vocabulary used in annotation of resources. In order to help this process of agreement we propose to use an initial controlled vocabulary that is shared by users during the annotation process. If users annotate resources using this controlled vocabulary, the consensus on the semantics of the annotation is given by construction. The issue rises when new terms (not present in the controlled vocabulary) are used to annotate resources, therefore, we need to define the process by which users reach a consensus on the semantics of these new terms (uncontrolled annotations) when used to annotate resources.

In order to aid the process of consensus on the use of semantics we need to focus on 2 parts of the life cycle of the annotation (Section 2):

1. Inserting the annotation: when users type the values of the annotation, a process of suggestion or auto completion of the value with existing terms in the vocabulary avoids the vocabulary problems presented in [18].

2. Ontology maturing via consensus. This can be done automatically, semi-automatically or manually by users.

In the following subsections we will describe in more detail the abovementioned processes.

4.3.1 Inserting the annotation

The aim at this level is to lower the effort needed by users to provide annotations, as a form of incentive, and also as a way of encouraging participation and foster consensus. We expect that users will provide more annotations if the annotation process is made very simple for them. Furthermore, the less the user types, the lower the probability for the user to make spelling mistakes [18] and the higher the likelihood the users will annotate more [36]. Some methods for lowering the effort required to annotate resources are:

Auto-complete with suggestions When users type the values of the annotation, a process of suggestion or auto completion of the value with existing terms in the vocabulary avoids the vocabulary problems presented in [18], namely errors in typing, basic level variation, and even possibly homonyms. Homonyms can be avoided by showing the different senses of the term with their respective meanings in some form, e.g., definition, glosses, or even by showing examples of resources annotated with the given sense (as done in [36]). In the later case, given that a sense is described with the annotated resources, when hovering over the resource, the system could show the other annotations of the particular resource. If the entered text cannot be found in the existing controlled vocabularies, we could try to look for the semantics of the entered text using external semantic knowledge bases such as Yago [41], sig.ma [24], DBPedia [2] among others, saving provenance information from this sources as proposed by [31]. This link with external resources will be further studied in [9]

Show community tags Showing the tags that have already been applied to the document, and letting users reuse them fosters annotation and lowers the effort needed to annotate resources. One possible method for showing community annotations is to use a variation of Tag clouds [25] that we will refer to as ”Semantic tag clouds”. The idea is to group tags not only considering the syntax (the actual text in the tag) but the semantics, therefore, if an item is tagged with two synonymous terms, they will be shown as only one item in the semantic tag cloud, being the size of the tag the sum of the usage of both terms for the given resource. We also propose the use of different font variation (e.g. color) to identify the annotations made by the user looking at the semantic tag cloud (provided that the user is authenticated) and those applied by other users. Also, different font variations could be used to identify the tags that are mapped to the controlled vocabulary (semantic tags or controlled annotations) and those whose meaning is still not clear (because they are in the form of uncontrolled annotations, or because the user still did not approve the suggested meaning for the tag, maybe automatically deduced, see next subsection 4.3.2).

The semantic tag clouds to be shown for each resource at annotation time could represent the currently applied annotations by all users for the given resource, grouped by type of attributes (tags, relations and others), i.e., different clouds for different attributes. Using semantic tag clouds users should have the possibility to *approve* (apply the tag) or *disapprove* the annotations made by other users. If the current user approves the annotation made by another user, then a new entry for that annotation is inserted in the database for the current user. If the user disapproves the annotation (by clicking a minus (-) icon or something similar next to the annotation), then the reliability of the selected annotation is decreased (similarly to [46] and [17]); the size of the annotation is also affected, and eventually if the reliability reaches 0 (zero) the tag will not be shown in the semantic tag cloud. The color of the rejected tag could also be changed (or crossed out) for the current user to denote that the particular tag has already been rejected.

Suggestion of annotations By analyzing the content and context of the resource being annotated the system could automatically suggest relevant annotations for the resource, giving the user the possibility to approve or disapprove these annotations, fostering convergence of tags and vocabulary [10]. This process is presented with the name of bootstrapping annotations in Section 3.

Figure 8: A possible user interface for a semantic tag cloud for this document.

4.3.2 Extracting concepts from uncontrolled annotations

Given that we have tried to achieve the most uniform tag application by using suggestions, auto-completion and possibly semantic tag clouds, but the user still applied terms that are not present in the controlled vocabulary, now we need to extract the meaning of these uncontrolled annotations and add the extracted concepts to the controlled vocabulary in the correct place.

The extraction of concepts from the annotations could be done in 3 different ways (not necessarily exclusive):

Automatically Several algorithms [28] [23] [17] (see Section 4.2) have already been proposed for extracting concepts and relations between the tags from free-text (uncontrolled) annotations over resources. If the extracted concept has a certainty over a given threshold, then this new concept could be automatically added to the controlled vocabulary.

Semi-automatically The results of automatic algorithms could be shown to the users that have applied the particular uncontrolled annotation, showing them the extracted concept and the relation of this concept with the existing controlled vocabulary, asking the user for this approval. Other mechanism for evaluating the accuracy of the computed concepts is to study the reuse pattern of this new concepts, e.g., if many users reuse this new concept, then it could mean that the concept is accurate, if not, it could be an indication that the concept is not well structured, or maybe is just not very relevant to the community. This issue will be further explored in deliverable 4.1.1 [43].

Manually If the user employs a free-text term to annotate a resource, the system could show in a particular color the tags that have not been mapped to the controlled vocabulary, and for which there is still no automatically extracted concept, indicating that the user could manually relate this particular term to the existing concepts in the controlled vocabulary. This issue will be further explored in deliverable 4.1.1 [43].

Manually extending the vocabulary by relating the new term to the existing controlled vocabulary is a matter of user interface design, but considering the collaborative nature of the present work, the user might not be fully aware of the possible effect of his/her manual change on the controlled vocabulary over other annotations[15]. Deliverable 2.2.2 [9] explores this problem and proposes an approach to detect these effects. The models presented in [42] and [17] (see Section 4.2) could be the base for the manual approach for extending the controlled vocabulary. The manual evolution of the underlying controlled vocabulary will be further explored in deliverable 4.1.1 [43] and 4.2.1 [44]. Each use case partner will have to develop its own methodology which best suits their own needs.

There are several proposed models for automatic extraction of concepts and relations based on free-text tags (uncontrolled annotations). The approaches presented in [17] and [28] (already mentioned in Section 4.2) contain surveys on the subject. Normal approaches include co-occurrence analysis, clustering techniques and others for extracting hierarchical taxonomies.

In [22] the authors present a consensus model in multi-person decision making. A model to select from different alternatives is presented, which helps people to decide from a set of ordered alternatives. This model could be useful if the (semi) automatic process for extracting semantics from uncontrolled annotations produces several alternatives, leaving the decision of the correct output (one or more) to the users. The model collects all the individual preferences, compares them, and then checks whether a consensus has been reached; in the positive case we could add the new concept to the controlled vocabulary, and in the negative case, another round of collecting the preferences has to be done. The model also defines two measures, one for *consensus measure*, which evaluates the level of agreement, and another for *proximity measure*, which evaluates the distance between an individual opinion from the *consensus measure*. The system could use this *proximity measure* to ask their preference only to those users that disagree the most, showing the current state of agreement, asking them if they would like to change their preferences, avoiding polling again all the users for their preferences.

4.4 Beyond the State of the Art

In this section we provided a review of related works showing that there is no single approach that includes all the features needed for dealing with the defined problem, and that rather multiple mechanisms should be taken into account in order to produce a solution that is easy to use and practically implementable

The novelty of our approach consists in recognizing that consensus reaching can be done also at annotation time, reducing the need for the creation of new terms whenever possible, therefore reducing the necessary effort to annotate resources. Semantic tag clouds is a new proposal that improves current tag clouds by considering the semantics of each annotation allowing users to approve or reject the annotations directly on the cloud, providing them information about which are semantically rich annotations and which are not. The consensus on the use of semantics of the annotations is achieved putting together in a unified model research from several areas such as ontology maturing, collaborative systems, natural language processing and consensus models for decision making.

This recognition of the need of multiple approaches at multiple levels of the life cycle of the annotation is an important improvement on the state of the art since current approaches seem not to consider all the dimensions needed to foster interoperability, allowing users to collaboratively evolve formally defined annotations in order to support semantics services.

5 Conclusion

The semantic web is slowed down by the lack of existing semantic data online, which limits the availability of semantic services to the end-users and thus does not motivate them to create new semantic content. This is a chicken and egg issue also known as the *cold start* problem. A solution to this would be to bootstrap the semantic annotation of resources online as soon as they are published.

In this deliverable, we first proposed a framework to help bootstrapping the annotation of local resources before they are published by combining diverse sources of implicit semantic and existing user and community knowledge. We propose to do this by combining existing information extraction techniques, machine learning techniques and novel research in merging and filtering these diverse sources of semantics.

We also foresee the issue of vocabulary explosion in a collaborative system, in addition to the cold start issue, once the users start annotating resources, if they all use a different uncontrolled vocabulary, the semantic services will not be able to work effectively. This can be seen as a controlled vocabulary cold start issue, as the users are not yet ready to use a semantically formalized set of concepts until this one include all the concepts they need to use. We propose a set of techniques to automatically enrich an existing controlled vocabulary from the set of uncontrolled annotations provided by the user. We also propose to solve part of this issue semi-automatically by providing smart interfaces to the users that will push them to reuse existing concepts to converge toward a common vocabulary.

References

[1] Karl Aberer, Philippe C. Mauroux, Aris M. Ouksel, Tiziana Catarci, Mohand S. Hacid, Arantza Illarramendi, Vipul Kashyap, Massimo Mecella, Eduardo Mena, Erich J. Neuhold, and Et. Emergent semantics principles and issues. In *Database Systems for Advances Applications (DASFAA 2004), Proceedings*, pages 25–38. Springer, March 2004.

[2] Sren Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: a nucleus for a web of open data. In *The Semantic Web*, pages 722–735. 2008.

[3] A Autayeu, F Giunchiglia, P Andrews, and Q Ju. Lightweight Parsing of Natural Language Metadata. *eprints.biblio.unitn.it*, 2009.

[4] Brian Auvine, , and Others. A manual for group facilitators., March 1977.

[5] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *The Scientific American*, 284(5):28–37, May 2001.

[6] Simone Braun, Andreas Schmidt, Andreas Walter, Gabor Nagypal, and Valentin Zacharias. Ontology maturing: a collaborative web 2.0 approach to ontology engineering. In *Proceedings of the Workshop on Social and Collaborative Construction of Structured Knowledge (CKC 2007) at the 16th International World Wide Web Conference (WWW2007)*, 2007.

[7] Tobias Burger, Olga Morozova, Ilya Zaihrayeu, Pierre Andrews, Juan Pane, and Borislav Popov. D2.1.2: Specification of models for representing single-user and community-based annotations of web resources. Technical report, UIBK, UNITN, ONTO, September 2009.

[8] Tobias Burger, Ilya Zaihrayeu, Pierre Andrews, Denys Babenko, Juan Pane, and Borislav Popov. Insemtives deliverable 2.1.1: Report on the state-of-the-art and requirements for annotation representation models. Technical report, UIBK, UNITN, ONTOTEXT, 2009.

[9] Tobias Burger, Ilya Zaihrayeu, Pierre Andrews, and Juan Pane. D2.2.2: Report on methods and algorithms for linking usergenerated semantic annotations to semantic web and supporting their evolution in time. Technical report, UIBK, UNITN, September 2009.

[10] Fabio Calefato, Domenico Gendarmi, and Filippo Lanubile. Towards social semantic suggestive tagging. In Giovanni Semeraro, Eugenio Di Sciascio, Christian Morbidoni, and Heiko Stoermer, editors, *SWAP*, volume 314 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

[11] Hsinchun Chen. Collaborative systems: Solving the vocabulary problem. *Computer*, 27(5):58–66, 1994.

[12] DJ Crandall, L Backstrom, and D Huttenlocher. Mapping the world's photos. *Proceedings of the 18th international conference on World wide Web 2009*, 2009.

[13] Marc Davis, Simon King, Nathan Good, and Risto Sarvas. From context to content: leveraging context to infer media metadata. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 188–195, New York, NY, USA, 2004. ACM.

[14] S Dill, N Eiron, D Gibson, D Gruhl, and R Guha. SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation. *Proceedings of the 12th international conference on World Wide Web*, 2003.

[15] Giorgos Flouris, Dimitris Plexousakis, and Grigoris Antoniou. Evolving ontology evolution. In *SOFSEM 2006: Theory and Practice of Computer Science*, pages 14–29. 2006.

[16] N Garg and I Weber. Personalized, interactive tag recommendation for flickr. *Proceedings of the 2008 ACM conference on Recommender Systems*, 2008.

[17] Domenico Gendarmi and Filippo Lanubile. Community-Driven ontology evolution based on folksonomies. In *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, pages 181–188. 2006.

[18] Scott Golder and Bernardo A. Huberman. The structure of collaborative tagging systems. *Journal of Information Science*, 32(2):198208, April 2006.

[19] Z Guan, J Bu, Q Mei, C Chen, and C Wang. Personalized tag recommendation using graph-based ranking on multi-type interrelated objects. *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 2009.

[20] B Hammond, A Sheth, K Kochut, and S Inc. Semantic enhancement engine: A modular document enhancement platform for semantic Applications over Heterogeneous Content. *Real World Semantic Web*, 2002.

[21] J Hendler. Web 3.0: Chicken farms on the semantic web. *COMPUTER-IEEE COMPUTER SOCIETY-*, 2008.

[22] E. Herrera-Viedma, F. Herrera, and F. Chiclana. A consensus model for multiperson decision making with different preference structures. *IEEE TRANSACTIONS ON SYSTEMS MAN AND CYBERNETICS PART A SYSTEMS AND HUMANS*, 32(3):392–402, 2002.

[23] Paul Heymann and Hector Garcia-Molina. Collaborative creation of communal hierarchical taxonomies in social tagging systems. Technical Report 2006-10, Stanford InfoLab, April 2006.

[24] http://www.deri.ie/. `http://sig.ma/`.

[25] Owen Kaser and Daniel Lemire. Tag-Cloud drawing: Algorithms for cloud visualization. *cs/0703109*, March 2007.

[26] R Krestel and P Fankhauser. Tag Recommendation using Probabilistic Topic Models. *ECML PKDD Discovery Challenge 2009 (DC09)*, 2009.

[27] DG Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 2004.

[28] Peter Mika. Ontologies are us: A unified model of social networks and semantics. In *International Semantic Web Conference*, pages 522–536, 2005.

[29] George A. Miller. WordNet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[30] Library of Congress. `http://www.loc.gov/index.html`. (last accessed on 01.06.2009).

[31] A Passant and P Laublet. Meaning of a tag: A collaborative approach to bridge the gap between tagging and linked data. *Proceedings of the WWW 2008 Workshop Linked Data on the Web (LDOW2008), Beijing, China, Apr*, 2008.

[32] B Popov, A Kiryakov, D Ognyanoff, and D Manov. KIMa semantic platform for information extraction and retrieval. *Natural Language Engineering*, 2004.

[33] M Rajman, P Andrews, MMP Almenta, and F Seydoux. Conceptual document indexing using a large scale semantic dictionary providing a concept hierarchy. In *Proc. of the 11th International Symposium on Applied Stochastic Models and Data Analysis*, pages 98–105, 2005.

[34] Corey L. Samuels and James A. Drake. Divergent perspectives on community convergence. *Trends in Ecology & Evolution*, 12(11):427–432, November 1997.

[35] Sandor P Schuman. Reaching consensus on consensus.

[36] Shilad Sen, Shyong K. Lam, Al Mamunur Rashid, Dan Cosley, Dan Frankowski, Jeremy Osterhouse, F. Maxwell Harper, and John Riedl. tagging, communities, vocabulary, evolution. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pages 181–190, Banff, Alberta, Canada, 2006. ACM.

[37] A Sheth, C Ramakrishnan, and C Thomas. Semantics for the semantic web: The implicit, the formal and the powerful. *International Journal on Semantic Web & Information*, 2005.

[38] Katharina Siorpaes, Mihail Konstantinov, and Borislav Popov. D3.1: Requirement analysis and architectural design of semantic content management platform. Technical report, UIBK, ONTO, September 2009.

[39] S Sood, K Hammond, S Owsley, and L Birnbaum. Tagassist: Automatic tag suggestion for blog posts. *infolab.northwestern.edu*, 2007.

[40] CAN Soules and GR Ganger. Connections: using context to enhance file search. *ACM SIGOPS Operating Systems Review*, 2005.

[41] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706, Banff, Alberta, Canada, 2007. ACM.

[42] V Tanasescu and O Streibel. Extreme tagging: Emergent semantics through the tagging of tags. In *Proceedings of the International Workshop on Emergent Semantics and Ontology Evolution*, 2007.

[43] tbd. *Insemtives Deliverable 4.1.1: Requirements and Design of a Generic Gaming Toolkit and API*, 2009.

[44] tbd. *Insemtives Deliverable 4.2.1: Human-driven annotation tools for Web services*, 2009.

[45] Thomas Vander Wal. Folksonomy: Coinage and definition. `http://www.vanderwal.net/folksonomy.html`.

[46] Jun Wang, Les Gasser, and Jim Houk. Convergence analysis for collective vocabulary development. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 1378–1380, Hakodate, Japan, 2006. ACM.

[47] Halb Wolfgang, Raimond Yves, and Hausenblas Michael. Building linked data for both humans and machines. In *LDOW08*, 2008.

[48] I. Zaihrayeu, L. Sun, F. Giunchiglia, W. Pan, Q. Ju, M. Chi, and X. Huang. From web directories to ontologies: Natural language processing challenges. In *6th International Semantic Web Conference (ISWC 2007)*. Springer, 2007.

[end of document]

